Efficiency of Entropy Coding as well as Dictionary based Technique for Lossless Data Compression

Pradeep S R [1],Prathibha S R [2], Monikashree T S [3]
[1,2,3] PG Student
pradeepsr09@gmail.com [1], prathi52@gmail.com [2], monikashree.ts@gmail.com
[1] Department of PG Studies, V T U, Gulbarga, Karnataka, India.
[2] Sri Siddartha Institute of Technology, Tumkur, Karnataka, India.
[3] Dr. Ambedkar Institute of Technology, Bangalore, Karnataka, India.

Abstract: Communication is exchange of information between the devices, which very important in today's life. Communication is wide utilized in all fields, such as telephone, television, weather forecasting and etc. There are different kinds of communication techniques. All the data which have to be transmitted has to be compressed in order to send more and more data. Communication is one of the main processes in many domains, but to send the information, the data has to be compressed, so that more and more data can be transmitted. In this paper presented the concept of data compression technique has been explained, which also includes efficiency of entropy and dictionary based techniques of lossless data type. And also include the merits and demerits, coding complexity, decoding capability and compression ratio of both entropy and dictionary based compression technique. Even it includes Huffman coding has been explained clearly using flowchart and implemented for a bit map image and encrypted and decrypted the image.

Keywords— Entropy, Huffman coding, Encryption, Decryption

I.INTRODUCTION

Data Compression is one of most important technique used in multimedia technology for transferring the video, audio or any text information from one place to another. Data compression algorithms are mainly used to reduce the number of bits required to represent the image or video or audio information. There is two parts in data compression i.e. compression and reconstruction. Based on reconstruction there are two algorithms i.e. lossless compression and lossy compression.
Lossless data compression allows exact data to be reconstructed from compressed data. Whereas the lossy data compression allows similar data to be reconstructed from compressed data but not identical as original.

This chapter involves the debate on the efficiency of entropy as well as dictionary based technique for lossless data compression, efficient usage of data types, merits and demerits of the both techniques, coding complexity, decoding capability, compression ratio.

1.1Efficient usage of the data types (text, images, etc.)
Entropy and Dictionary based techniques are two types under the lossless data compression. Huffman coding is an entropy type data compression technique. Huffman coding is based on the frequency of occurrence of a data item i.e. pixel in images. The technique is to use a lower number of bits to encode the data in to binary codes that occur more frequently. It works better for JPEG image format. LZW is a dictionary based type data compression technique. LZW replaces strings of characters with single code. LZW algorithm works best for files containing lots of repetitive data. This is suited for text and monochrome images like Tagged Image File Format (TIFF) and Graphic Interface Format (GIF). It is not suited for an image that doesn‟t have repetitive data [1].

1.2 Merits and Demerits of two techniques.

1.2.1 Merits and Demerits of Entropy coding.

Huffman coding is one of the most popular compression methods of entropy coding, which translate fixed-size pieces of input data into variable-length symbols. Huffman coding is widely used of its simplicity, high speed and lack of difficulty. Produces lossless compressions of images [1].
All the codes of encoded data are of different size, therefore it is difficult to for decoder to know whether last of bit of code has reached or not. Thus encoded output mat be corrupted and final image may not be same as original, hence need to send Huffman table at beginning of compressed file hence it is slow in compression [1].

Copyright 2014- IJRDT

1.2.2 Merits and Demerits of Dictionary based coding
LZW is one type of Dictionary based algorithm which replaces the strings of characters with single codes. LZW compression is the best technique for reducing the size of files containing more repetitive data. LZW compression is fast and simple to apply. Since this is a lossless compression technique, none of the contents in the file are lost during or after compression. And it need not pass the dictionary table to decompression code [1].
LZW is suitable of the image having the repetitive data, but if the image contains different data, then size of compressed file by the LZW will be larger. Hence the LZW is not suitable for the images of different data. It is very expensive to use [1].

1.3 Coding complexity
The term coding complexity refers to the ability of each lossless data compression techniques, the coding complexity for the entropy method is less as compared to the dictionary algorithm, however the technique which has more complexity is the arithmetic coding than the Huffman coding in entropy method. The one major disadvantage in lossless data compression is that by increasing the sizes of the search buffer and the look-ahead buffer will resolve the problems. A close look, reveals that it also leads to increases in the number of bits required to encode the offset and matched the corresponding string as well as increasing the length and increasing the processing complexity. Hence the there is less coding complexity in the entropy type compared to the dictionary algorithms. Therefore the complexity of a string is the length of the string's shortest description is fixed [2].

1.4 Decoding capability
Decoding mainly refers to reduction of the probability of errors by repetition of messages in transmission by using redundancy. Decoding capability holds good for the dictionary type of algorithm as they offer good compression and decompression for the data [2]. When compared with the encoding algorithm the decoding becomes difficult as the code length present in entropy methods is different. Since the decoder knows the rule applied in the encoding, it can reconstruct the dictionary and decode the input text stream from the received string. Whereas there is no look ahead possible in entropy method making the method less capable for decoding. Decoding makes the job simpler as there is no need for string comparison required in dictionary type of algorithms [3].

1.5 Compression Ratio
Data compression techniques are specifically dependent on the type of data that has to be compressed and on the desired performance. Typical measures of performances are time and space complexity, compression ratio is used to quantify the reduction in data-representation size produced by a data compression algorithms. Compression ratio is the main constraint being involved during comparison of each technique"s efficiency. The compression ratio is basically given as the ratio of compressed size versus the uncompressed size of data may be for an image or data it is represented the same [4]. As discussed before about the decoding capacity the efficiency of the dictionary type technique increases as there is good compression and decompression possible utilizing less
memory space. While considering the entropy type of algorithms like Huffman coding and arithmetic coding the RAM space is more hence this in turn implies that the compression ratio is also high for these techniques. The dictionary based algorithms have less compression ratio and are used in image compression like Graphic Interface Format (GIF) [4].

II.PROPOSED WORK

Data compression is a technique which reduces the number of bits to transmit and represent the required information i.e. video, audio, data signal and etc. There are two techniques in data compression i.e. lossless and lossy data compression.

Huffman coding is type of lossless data compression technique. This compresses the required information and reconstructs information exactly as original. Using this technique the codes are generated which is known as Huffman codes and these codes are prefix codes which is best for the particular image or model. Huffman technique is based on the two observations.

- The symbols that occur more frequently have less codeword rather than the symbols that occur less frequently.
- The same symbol that occurs less frequently has the same length of codeword.

These code words of particular probability of color are replaced in image, in place of pixels of a particular color. Initially the bits require to represent the color will be more, but by using this technique less number of bits will be required to represent the color and transmit.

A) *Flow Chart to describe the logic of Huffman coding*

The figure 1 shows the flow chart of encryption part of the Huffman coding for a bit map image. Here initially image is either can be created or directly read the created image, then the probability of each colour is calculated by ratio of total number of pixels of particular color to the total number of pixels of whole image. Then the probabilities of all color are assigned to the variable i.e. „p". Then while condition is considered to check and repeat the while loop until the length of „p" is greater than 1. Firstly the length of „p" is usually greater than one, hence it satisfies the condition and enters the loop. Then the probabilities in „p" is arranged in descending order and sums the least two probabilities. Then

Copyright 2014- IJRDT

the length of „p" is reduced by 1 and assigns the sum value to least place in „p". Again the loop repeats until the length of „p" becomes equal to 2. The code words for each probability is generated, logic for the code word generation is mentioned in code, which is in appendix. Then these generated code words of particular color probability are replaced in the each pixel value of that particular color. Then the both encrypted image and code word table is transmitted to the receiver part.

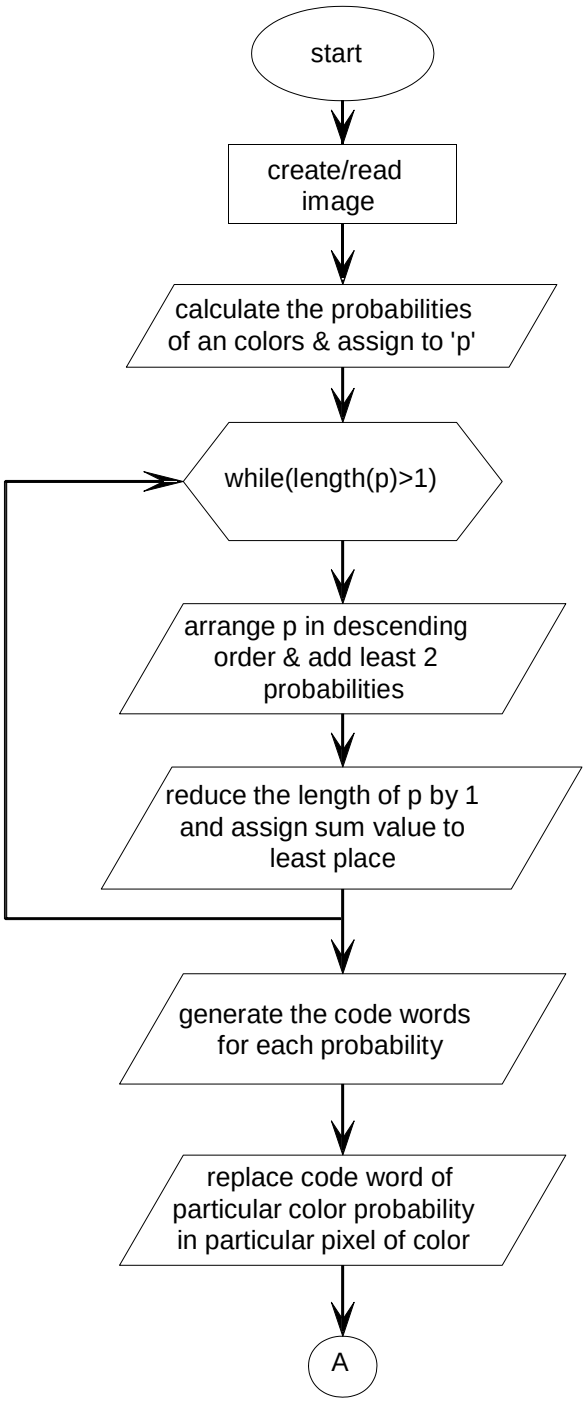

Fig.1: Flow chart of encrypted part

The figure 2 illustrates the flowchart of the decryption on image in Huffman coding. The code word table and encrypted image sent by the transmitter is received and replaces the original color intensity value with respect to the particular code word of color probability.

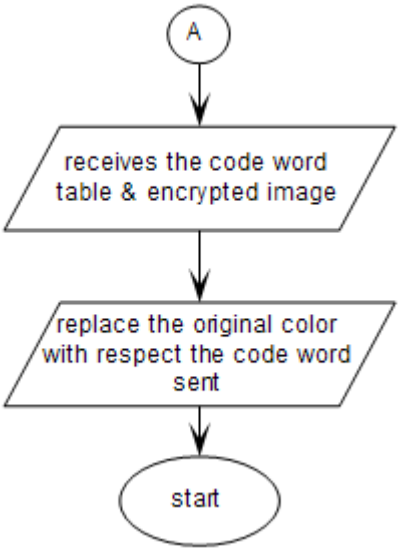

Fig.2: Flow chart of decryption of the image

III.RESULT ANALYSIS

Fig.3: Output of Original Image

red_prob = 0.2400

grn_prob = 0.1600

blu_prob = 0.3600

blk_prob = 0.2400

The figure 3 illustrates the original image, which has four colors i.e. red, green, blue and black. Huffman coding procedure is applied for above image. Initially the total probability of each color is calculated, which is calculated

Copyright 2014- IJRDT

by dividing the total number of pixels of each color by the total number of pixels in a whole image. The probability values of colors of above image are shown above.

pp1 = 0.3600 0.2400 0.2400 0.1600

pp2 = 0.4000 0.3600 0.2400

pp3 = 0.6000 0.4000

All the probabilities are assigned to a variable and arranged in descending order. And least two probabilities of variable are added and length of variable is reduced by 1 and assigns the all probabilities to new variable, then assigns the sum value in last place of new variable. And this is repeated until the length of new variable is equal to two, as shown above.

s1 = 11 = blue

s2 = 10 = red

s3 = 01 = black

s4 = 00 = green

Fig.4: Output of Encrypted Image

The symbol for each probability is generated which is shown above i.e. s1, s2, s3, s4. These symbols are known as code words which are replaced in the each pixel value of that particular color. Then a resultant image is known as the encrypted image which is shown in figure 4.

Entropy = $-\sum_{k=1}^{length(prob)} prob\ k\ log2 .* prob(k)$

= 1.9419

Average length = Prob. of symbol x No. of bits to represent symbol

= 2

Redundancy = (Entropy-Average length)*100

= 5.8099

Efficiency = 100-redundancy= 94 %

The figure 5 illustrates the decryption of the encrypted image. The code words which have been replaced in the image have been replaced by the original intensity value and reconstruct the original image as shown image in figure 5.

Fig.5: Output of Decrypted Image

IV.APPLICATIONS

1. Suppression of zero's in a file **(**Zero Length Suppression).
2. Silence in audio data, or pauses in conversation etc**.**
3. Bitmaps
4. Blanks in text or program source files
5. Backgrounds in images
6. Other regular image or data tokens

V.ADVANTAGES & DISADVANTAGES

ADVANTAGES

- ✓ Algorithm is easy to implement
- ✓ Produce a lossless compression of images

DISADVANTAGES

- ✓ Efficiency depends on the accuracy of the statistical model used and a type of image.
- ✓ Algorithm varies with different formats, but few get any better than 8:1 compression.

VI.CONCLUSION

Lossless data compression technique is better than the lossy data, since it allows reconstructing the image or information similar to the original information. Entropy technique is slow compare to the dictionary based technique since it has to send the symbol table to receiver part.

Huffman coding is type of lossless data compression technique which generates the code words and replaces those code words in the original image in order to compress

Copyright 2014- IJRDT

the image and reduce the size of the image to transmit. It is used in JPEG compression. It produces optimal and compact code but relatively slow since it has to send code table also.

REFERENCE

[1] 'Compression Using Huffman Coding',Mamta Sharma, S.L. Bawa D.A.V. college.
[2] Pu, I.M., 2006, Fundamental Data Compression, Elsevier, Britain.

[3] T.C.Bell, J.G. Cleary, and I.H. Written., Text Compression, Advanced Reference Series. Prentice Hall, Englewood Cliffs, NJ, 1990.

[4] Lossless Data Compression. Recommendation for Space Data Systems Standards, CCSDS 121.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, May 1997.
[5] S.R. KODITUWAKKU, U. S.AMARASINGHE, "Comparison Of Lossless Data Compression Algorithms For Text Data" Indian Journal of Computer Science and Engineering, Vol 1 No 4 416-425.pp 416 -425

Copyright 2014- IJRDT